PERANCANGAN PROGRAM PENGHITUNG JUMLAH KENDARAAN DI LINTASAN JALAN RAYA SATU ARAH MENGGUNAKAN BAHASA PEMROGRAMAN C++ DENGAN PUSTAKA OPENCV

Publikasi Jurnal Skripsi

Disusun Oleh :

FAJAR MIT CAHYANA

NIM : 0610630037-63

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2014

	KEMENTRIAN PENDIDIKAN DAN KEBUDAYAAN UNIVERSITAS BRAWIJAYA FAKULTAS TEKNIK JURUSAN TEKNIK ELEKTRO Jalan MT Haryono 167 Telp & Fax. 0341 554166 Malang 65145	**KODE** **PJ-01**

PENGESAHAN

PUBLIKASI HASIL PENELITIAN SKRIPSI

JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK UNIVERSITAS BRAWIJAYA

NAMA : FAJAR MIT CAHYANA

NIM : 0610630037

PROGRAM STUDI : TEKNIK ELEKTRONIKA

JUDUL SKRIPSI : PERANCANGAN PROGRAM PENGHITUNG JUMLAH KENDARAAN DI LINTASAN JALAN RAYA SATU ARAH MENGGUNAKAN BAHASA PEMROGRAMAN C++ DENGAN PUSTAKA OPENCV

TELAH DI-REVIEW DAN DISETUJUI ISINYA OLEH:

Pembimbing I	Pembimbing II
M. Julius St., Ir., MS	**R. Arief Setiawan ST., MT**
NIP. 19540720 198203 1 003	**NIP. 19750819 199903 1 002**

IMPLEMENTASI *INVERS KINEMATICS* PADA SISTEM PERGERAKAN *MOBILE ROBOT* RODA MEKANUM

Fajar Mit Cahyana[1], M. Julius[2], R. Arief Setiawan[3]
[1]Mahasiswa Teknik Elektro UB, [2,3]Dosen Teknik Elektro UB
hendrayawanveri@gmail.com

Abstrak **- OpenCV adalah sebuah library yang berisi fungsi-fungsi pemograman untuk teknologi computer vision secara real time. OpenCV sudah menggunakan antarmuka bahasa C++ dan seluruh pengembangannya terdapat dalam format bahasa C++. Contoh aplikasi dari OpenCV yaitu interaksi manusia dan komputer; identifikasi, segmentasi dan pengenalan objek, pengenalan wajah, pengenalan gerakan dan penelusuran gerakan.**

Skripsi ini membahas perancangan program program penghitung jumlah kendaraan di lintasan jalan raya satu arah menggunakan bahasa pemrograman c++ dengan pustaka opencv 2.4.4. Video yang digunakan sebagai objek penelitian dipisahkan background dan foregroundnya dengan mengguakan background subtraction. Foreground dicari posisi titik centroidnya dalam frame kemudian dihitung berapa titik centroid yang melewati garis yang ditetapkan dalam program yang dirancang.

I. Pendahuluan

1.1. Latar Belakang

Untuk menentukan kepadatan rata-rata lalu-lintas diperlukan adanya survey penghitungan jumlah kendaraan yang melintas di jalan raya. Pelaksannan survey tersebut biasanya dilakukan oleh seorang pengamat yang dimungkinkan terjadinya *human error* dalam proses penghitungan karena terlalu padatnya jumlah kendaraan yang lewat, pengaruh lingkungan atau kondisi internal peneliti itu sendiri sehingga mengakibatkan kurang akuratnya proses penghitungan yang dilakukan langsung oleh seorang peneliti. Selain rentan terjadinya *human error*, penghitungan yang dilakukan oleh manusia memerlukan biaya tersendiri untuk setiap pelaksanaanya sehingga kurang efisien.

Oleh karena itu, penulis berusaha melakukan sebuah perancangan dan analisis program penghitung jumlah kendaraan di lintasan jalan raya menggunakan bahasa pemrograman C++ dengan pustaka OpenCV. OpenCV digunakan karena memiliki banyak subprogram atau *library* yang dapat dikombinasikan sehingga memiliki berbagai fungsi dalam pemrograman yang berkaitan dengan pengolahan citra digital. Sedangkan bahasa pemrograman C++ digunakan karena Pustaka OpenCV 2.4.4 hanya dapat dijalankan dengan menggunakan bahasa pemrograman C dan C++.

1.2 Ruang Lingkup

Ruang lingkup pada skripsi ini ditekankan pada :

1. Bagaimana penggunaan pustaka OpenCV dengan bahasa pemrograman C++ untuk proses pemrosesan citra digital dalam penelitian ini.
2. Bagaimana proses pemisahan latar belakang yang bersifat statis dengan objek yang bergerak pada video.
3. Bagaimana mengunci objek yang bergerak pada video yang diamati.
4. Bagaimana menghitung objek yang bergerak pada video yang diamati.

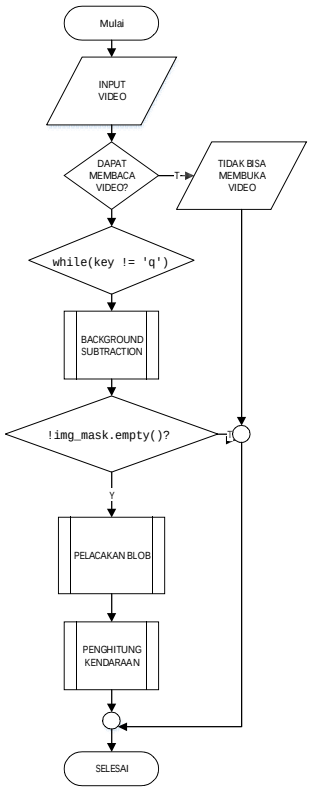
II. Perancangan

2.1 Perancangan Secara Umum

Perancangan sistem secara umum merupakan tahap awal sebagai acuan dalam perancangan sistem yang akan dibuat. Pada bagian ini akan dibahas mengenai garis besar dari masing-masing bagian dari sub-program dan proses apa yang akan terjadi dari tiap-tiap bagiannya.

Perancangan program terbagi menjadi empat bagian yang terdiri dari inisialisasi masukan video, *background subtraction*, pelacakan *blob* (*blobtracking)* dan algoritma penghitung jumlah kendaraan.

Perancangan program penghitung kendaraan ditunjukkan dalam Gambar 1.



Gambar 1 Diagram blok program.

2.2 Perancangan *Backround substraction*

Dalam perancangan ini data masukan berasal dari video dari kamera yang berisi kumpulan informasi biner yang memuat intensitas piksel yang ada di dalamnya. Sedangkan output dari *background subtraction* berupa gambar *foreground* yang nantinya akan diproses dalam proses *blobtracking.*

2.3 Perancangan *Blobtracking*

Pada perancangan ini pustaka *cvblob* digunakan untuk mempermudah programmer dalam memisahkan tiap-tiap gumpalan yang terdapat dalam suatu gambar dan kemudian melabelinya sehingga antara gumpalan yang satu dengan gumpalan yang lainnya dapat dibedakan dengan identitas yang berbeda.

2.4 Penghitung Kendaraan

Dalam proses penghitungan kendaraan ini dibagi menjadi empat bagian yaitu pengaturan metode penghitungan, pengaturan garis hitung, proses penghitungan dan proses penandaan jalur (*track).*

2.4.1 Pengaturan Garis Hitung

Pengaturan garis hitung dilakukan untuk mendapatkan garis hitung yang paling sesuai dengan kondisi jalan yang akan diamati. Pengaturan garis hitung dilakukan dengan melakukan dua kali *mouse click* sehingga membentuk garis tegak lurus dengan arah objek yang akan dihitung.

2.4.2 Perancangan Metode Penghitungan Kendaraan

Pada perancangan metode penghitungan kendaraan, *blob* hasil dari proses *blobtracking* dibandingkan posisi titik pusatnya dengan garis hitung yang telah ditetapkan. Setiap titik pusat melewati garis yang telah ditetapkan maka akan menambah jumlah kendaraan yang berhasil dihitung.

III. PENGUJIAN DAN ANALISIS

3.1 Prosedur Operasional Program

Pada awal program dijalankan dilakukan pengaturan garis yang dilintasi objek yang akan dihitung jumlahnya. Jika garis yang ditetapkan berfungsi, maka garis hitung akan ditampilkan dalam jendela program. Pada awal proses ditampilkan gambar masukan dari sumber dan pesan "Tentukan Garis Hitung". Kemudian pengguna dapat melakukan pengaturan garis dengan melakukan klik pada mouse pada kedua titik ujung garis hitung. Tampilan dari proses pengaturan garis hitung ditunjukkan dalam Gambar 2.

Gambar 2. Tampilan Awal Penentuan Garis Hitung

Pengaturan garis hitung ditunjukkan dalam Gambar 3.

Gambar 3. Pengaturan Garis Hitung

Tampilan program jika garis hitung terdeteksi ditunjukkan dalam Gambar 4.

Gambar 4. Tampilan Program Jika Garis Hitung Terdeteksi

Berikutnya program akan mendeteksi objek dengan menampilkan beberapa jendela program antara lain Jendela input, *frame difference*, *Blob Tracking* dan Penghitung Kendaraan. Jendela *input* ditunjukkan dalam Gambar 5.

Gambar 5. Jendela *Input*

Jendela *frame difference* ditunjukkan dalam Gambar 6.

Gambar 6. Jendela *Frame Difference*

Jendela *blobtracking* ditunjukkan dalam Gambar 7.

Gambar 7 *Blob Tracking*

Jendela penghitung kendaraan ditunjukkan dalam Gambar 8.

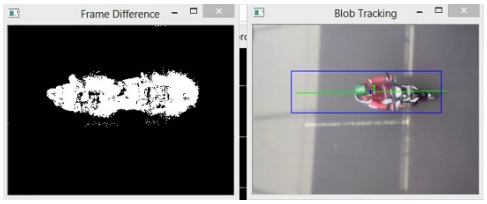
Gambar 8. Penghitung Kendaraan

3.2 Pengujian

Pengujian pada skripsi ini terdiri dari dua bagian meliputi pengujian *Blob tracking* dan Pengujian penghitung kendaraan Pengujian dilakukan terhadap dua buah video yang memuat memiliki sudut pengambilan gambar yang berbeda, yaitu 45^o dan 90^o.

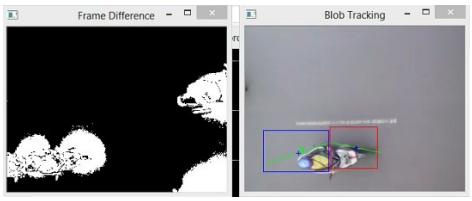
3.2.1 Pengujian *Blob Tracking*.

Pengujian *blobtracking* dikatakan berhasil apabila setiap kendaraan yang melintas dideteksi oleh program sebagai satu buah objek. Kemungkinan lain yang akan terjadi adalah sebuah objek dideteksi menjadi lebih dari satu objek. Satu objek yang terdeteksi sebagai satu *blob* ditunjukkan dalam Gambar 9.



Gambar 9. Satu Objek yang Terdeteksi Sebagai Satu *Blob*

Satu objek yang terdeteksi sebagai lebih dari satu *blob* ditunjukkan dalam Gambar 10.



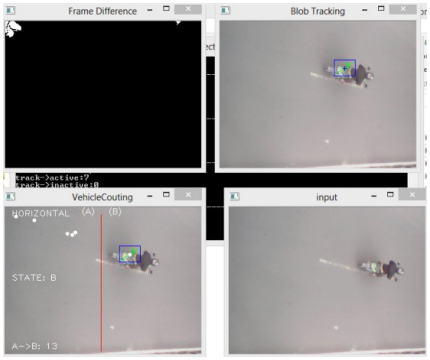
Gambar 10. Satu Objek yang Terdeteksi Sebagai Satu *Blob*

3.2.2 Pengujian Penghitung Kendaraan

Dalam pengujian ini dihitung kendaraan yang melintas menggunakan program penghitung kendaraan dan dengan penghitungan manual. Hasil yang didapatkan dibandingkan untuk mendapatkan ketepatan program dalam menghitung jumlah kendaraan.

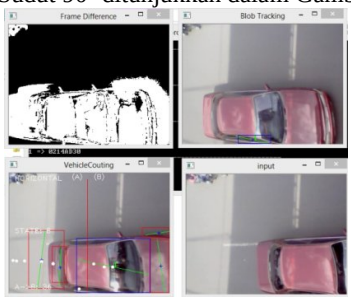
Pengujian dilakukan dengan menggunakan dua buah video. Video yang pertama adalah hasil pengambilan video dengan sudut 45^o dan video kedua yang digunakan adalah hasil

pengambilan video dengan sudut 90^o. Tampilan Pengujian Program Penghitung Kendaraan Dengan Sudut 45^o ditunjukkan dalam Gambar 11.



Gambar 11. Pengujian Program Penghitung Kendaraan Dengan Sudut 45^o

Pengujian Program Penghitung Kendaraan Dengan Sudut 90^o ditunjukkan dalam Gambar 12.



Gambar 12. Pengujian Program Penghitung Kendaraan Dengan Sudut 90^o

Hasil penghitungan dalam Pengujian Program Penghitung Kendaraan Dengan Sudut 45^o menunjukkan sebanyak 90 kendaraan berhasil dihitung dari 120 kendaraan yang melewati jalur tersebut. Keberhasilan prpgram tersebut dalam menghitung adalah sebesar 75 %.

Hasil penghitungan dalam Pengujian Program Penghitung Kendaraan Dengan Sudut 90^o menunjukkan sebanyak 91 kendaraan berhasil dihitung dari 129 kendaraan yang melewati jalur tersebut. Keberhasilan program tersebut dalam menghitung adalah sebesar 70 %.

3.3 Analisis Sistem

Analisis sistem dilakukan berdasar pada bagian tinjauan pustaka. Analisis dilakukan terhadap parameter-parameter sistem yang digunakan dan hasil yang didapat dari proses pengujian.

3.3.1 Analisis Hasil Pengujian *Blobtracking*

Dalam proses *blobtracking* terdapat satu objek yang dideteksi sebagai dua objek. Hal tersebut dapat terjadi karena hasil dari substraksi *background* yang membentuk dua gumpalan pada *foreground*. Dua gumpalan tersebut terpisah karena adanya proses erosi dan dilasi yang diterapkan pada objek, di mana tujuan awal dari dilakukannya erosi dan dilasi adalah untuk memisahkan objek yang satu dengan objek yang lainnya.

Keberhasilan program dalam menentukan *foreground* dipengaruhi juga dengan kecepatan objek yang diproses. Semakin cepat objek akan menghasilkan *foreground* yang semakin panjang dan akan terpisah menjadi dua objek jika terlalu cepat. Hal tersebut dapat terjadi karena *foreground* yang terbentuk menjadi pecah dan tidak menyatu sehingga dideteksi sebagai *blob* yang berbeda.

Pada kecepatan rendah *blob* dapat dengan mudah dideteksi secara tepat karena *foreground* yang dihasilkan utuh. Bentuk *foreground* yang utuh tidak akan bisa terpisah oleh proses erosi dan dilasi yang diterapkan pada objek.

3.3.2 Analisis Hasil Penghitungan Kendaraan

Dalam video yang diambil dalam sudut 45^0 memiliki nilai prosentase ketepatan hitung yang lebih besar. Hal tersebut dikarenakan luasan daerah yang tertangkap oleh kamera lebih luas, jumlah *frame* yang diproses lebih banyak sehingga lebih mudah dideteksi.

Besar sudut pengambilan gambar juga berpengaruh kepada besarnya *blob* yang didapatkan. Jika pengambilan gambar dilakukan secara tegak lurus maka objek akan terlihat lebih besar. Objek yang terlalu besar pada gambar tidak akan terdeteksi sebagai suatu *blob* karena ketika objek tersebut lebih besar daripada luasan *frame* maka tidak akan membentuk suatu objek.

IV. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Kesimpulan yang dihasilkan dalam penelitian ini adalah:

1. Untuk melakukan proses *background subtraction* untuk pengujuan secara real time secara dapat menggunakan metode *frame difference* dengan kecepatan 483,33 *frame* per detik.
2. Untuk mendapatkan hasil objek yang dapat dideteksi melalui *blobtracking* perlu ditetapkan *filter blob* yang sesuai dengan kriteria objek yang akan dideteksi.
3. Untuk menggunakan program penghitung kendaraan harus memperhatikan kecepatan objek yang akan dihitung dan pengambilan sudut kamera yang tepat. Sudut optimal pengambilan gambar terletak pada sudut 60^0 dengan tingkat keberhasilan 79%

4.2 Saran

Saran untuk memperoleh hasil penelitian yang lebih baik adalah:

1. Menggunakan perangkat *hardware* berupa kamera yang memiliki sensitifitas cahaya yang baik sehingga objek dapat terdeteksi secara akurat.
2. Menggunakan metode pemisahan latar belakang dan objek yang menghasilkan latar belakang dengan nilai *pixel* yang lebih stabil sehingga latar belakang yang didapatkan benar-benar bersifat statis.
3. Menggunakan *frame* berukuran sekecil mungkin dengan area yang memadahi untuk digunakan dalam proses penghitungan kendaraan sehingga menghemat ruang *memory* yang degunakan selama proses untuk mendapatkan proses penghitungan yang cepat dan akurat.
4. Menggunakan posisi yang lebih tinggi untuk mendapatkan area pandang kamera yang memuat semua objek yang akan dihitung.

Daftar Pustaka

Arymurthy, Aniati Murni. 1992. *Pengantar Pengolahan Citra.* Jakarta: PT. Elex Media Komputindo.

Bradsky, Gary. 2008. *Learning OpenCV*. California: O'Reilly Media.

Fairhurst, Michael and Smith, Stephen L. and Mitchell, John . 1995. "Automated Image-Analysis in Visuo-Motor Testing for the Specification of an Integrated Evaluation and Terapy Suport Toll for Rehabilitation". IEEE Transactions on Rehabilitation Engineering.

Fu Chang, Chun Jen Chen, and Chi Jen Lu. 2003. "A Linear Time Component-Labelling Algorithm Using Contour Tracing Technique". Taipei: Institute of Information Science, Academia Sinica.

Hinz, S. 2005. "Fast and Subpixel Precise Blob Detection and Attribution". IEEE International conference of Imagee Processing. Volume 3.

Kumar Jain, Anil. 1989. *Fundamentals of Digital Image Processing*. New Jarsey: Prentice Hall Inc.

Laganiere, Robert. 2009. *OpenCV 2 Computer Vision Application and Programmng Cookbook*. Birmingham: Pact Publishing.

Lindenberg, T. 1993. "Detecting Salient Blob-Like Structures and Their Scales with A Scale-Space Primal Sketch: A Method for Focus of Attention". International Journal of Computer Vision. Volume 11.

Low, Adrian. 1991. *Introductory Computer Vision and Image Processing Paperback.* Mcgraw Hill Book Co Ltd.

Rosenfeld, A and C.Y. Sher. 1998. "Detecting Images Primitives UsingFeature Pyramids" Information Sciences: An International Journal. Volume 107.

Serra, J. 1983. *Image Analysis and Mathematical Morphology*, New York: Academic Press.

Shapiro, Linda. 2001. *Computer Vision.* Washington: The University of Washington.

Yang Q. and B. Parvin. 2002. "Chef: Convex Hull of Elliptic Features for 3D Blob Detection".